APPEAL BRIEF UNDER 37 C.F.R. § 41.37

TABLE OF CONTENTS

Page

1. REAL PARTY IN INTEREST 2
2. RELATED APPEALS AND INTERFERENCES 3
3. STATUS OF THE CLAIMS 4
4. STATUS OF AMENDMENTS 5
5. SUMMARY OF CLAIMED SUBJECT MATTER 6
6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL 12
7. ARGUMENT 13
8. CLAIMS APPENDIX 26
9. EVIDENCE APPENDIX 36
10. RELATED PROCEEDINGS APPENDIX 37

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Thomas Vogt et al.	Examiner: Thomas J. Dailey
Serial No.: 10/662,125	Group Art Unit: 2152
Filed: September 12, 2003	Docket: 2058.226US1
For: Dynamic access of data	

APPEAL BRIEF UNDER 37 CFR § 41.37

Mail Stop Appeal Brief- Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

The Appeal Brief is presented in support of the Notice of Appeal to the Board of Patent Appeals and Interferences, filed on October 31, 2008, from the Final Rejection of claims 1-50 of the above-identified application, as set forth in the Final Office Action mailed on July 23, 2008.

The Commissioner of Patents and Trademarks is hereby authorized to charge Deposit Account No. 19-0743 in the amount of $540.00 which represents the requisite fee set forth in 37 C.F.R. § 41.20(b)(2). The Appellants respectfully request consideration and reversal of the Examiner's rejections of pending claims.

1. REAL PARTY IN INTEREST

The real party in interest of the above-captioned patent application is the assignee, SAP AG.

2. RELATED APPEALS AND INTERFERENCES

An appeal is pending with respect to patent application no. 10/365,672.

3. STATUS OF THE CLAIMS

The present application was filed on September 12, 2003 with claims 1-60. A non-final Office Action was mailed on January 10, 2008. A Final Office Action was mailed July 23, 2008. Claims 1-50 stand twice rejected, remain pending, and is the subject of the present Appeal.

4. STATUS OF AMENDMENTS

No amendments have been made subsequent to the Final Office Action dated July 23, 2008.

5. SUMMARY OF CLAIMED SUBJECT MATTER

Aspects of the present inventive subject matter include, but are not limited to, a heterogeneous information technology system in which compatible and incompatible client systems are able to dynamically access master data stored in a master database maintained by a master data server.

INDEPENDENT CLAIM 1

Some embodiments of the Application are related to a system, comprising:

a master data server, to maintain a master database storing master data objects, the master data server using master identifiers to identify the master data objects, the master database being accessible to clients (Figure 1 reference numerals 100, 116, 120, 122, 124; [0006] page 2, [0068] page 11, [0078] page 14, [0079] page 15); and

an integration server, in response to a request from a client to access master data identified by a client identifier, to map the client identifier to a master identifier, retrieve a master data object from the master database based on the master identifier, and map the master data object to a mapped data object based on a set of mapping rules associated with the client (Figure 1 reference numeral 114, Figure 3, reference numeral 102, [0006] page 2, [0081] page 15).

INDEPENDENT CLAIM 19

Some embodiments of the Application are related to a system, comprising:

a master data server, to maintain a master database storing master data objects, each object having a set of attributes, the master database being accessible to clients, each client processing a subset of attributes of the master data objects (Figure 1, reference numerals 100, 116, 120, 122, 124, [0006] page 2, [0068] page 11, [0078] page 14, [0079] page 15); and

an integration server, in response to a request from any one of the clients to access a master data object, to retrieve the master data object from the master database and map the master data object to a mapped data object based on a set of mapping rules associated with the client so that the mapped data object contains the subset of attributes in a format that can be

processed by the client (Figure 1 reference numeral 114, Figure 3, reference numeral 102, [0006] page 2, [0081] page 15).

INDEPENDENT CLAIM 20

Some embodiments of the Application are related to a method, comprising:

maintaining a master database at a data server, the master database containing master data objects, the master database accessible to clients (Figure 1 reference numerals 100, 116, 120, 122, 124; [0006] page 2, [0068] page 11, [0078] page 14, [0079] page 15);

receiving a request from a client to access master data, the request containing a client identifier ([0006] page 2);

mapping the client identifier to a master identifier ([0006] page 2);

retrieving a master data object based on the master identifier ([0006] page 2);

mapping the master data object to a mapped data object based on a set of mapping rules associated with the client ([0006] page 2); and

sending the mapped data object to the client ([0005] page 2).

INDEPENDENT CLAIM 28

Some embodiments of the Application are related to a method for maintaining data comprising:

providing a master database having master data shared by at least two clients (Figure 1, reference numeral 116, [00035] page 6);

providing an interface for updating the master database ([00035] page 6);

providing an interface for mapping subsets of the master data into mapped data having a format that is acceptable to each client ([00035] page 6); and

providing a user interface for entering and displaying subsets of the master data ([00035] page 6).

INDEPENDENT CLAIM 35

Some embodiments of the Application are related to a method for maintaining data, comprising:

receiving a first identifier used by a first client to identify a data object, and a request to delete the data object, the data object being stored in a database maintained by a data server, the database being accessible to the first client and a second client ([0043] page 7);

mapping the first identifier to a second identifier used by the second client to identify the data object ([0043] page 7);

mapping the first identifier to a third identifier used by the data server to identify the data object ([0043] page 7);

querying the second client based on the second identifier to determine whether the second client is using the data object ([0043] page 7); and

if the second client is not using the data object, deleting the data object from the database based on the third identifier ([0043] page 7).

INDEPENDENT CLAIM 39

Some embodiments of the Application are related to a method comprising:

receiving a first set of communications from a first client ([0048] page 8);

analyzing the first set of communications to find a set of characteristics that the first client associates with a data object used in the first set of communications ([0048] page 8);

analyzing other communications received from clients to find additional sets of characteristics that clients associate with data objects that have the same characteristics as the first set of characteristics ([0048] page 8);

placing the first client and clients who sent a set of characteristics that are the same as the first set of characteristics into a client group ([0048] page 8); and

generating a data distribution path to allow updates of the set of characteristics to be sent to the client group ([0048] page 8).

INDEPENDENT CLAIM 40

Some embodiments of the Application are related to a computer program product, tangibly stored on a machine-readable medium, for dynamic access of master data, comprising instructions operable to cause a programmable processor to:

maintain a master database (Figure 1, reference numeral 116) at a data server (Figure 1, reference numeral 100), the master database containing master data objects, the master database accessible to clients ([0049] page 8);

receive a request from a client to access master data, the request containing a client identifier ([0049] page 8);

map the client identifier to a master identifier ([0049] page 8);

retrieve a master data object based on the master identifier ([0049] page 8);

map the master data object to a mapped data object based on a set of mapping rules associated with the client ([0049] page 8); and

send the mapped data object to the client ([0049] page 8).

INDEPENDENT CLAIM 41

Some embodiments of the Application are related to a computer program product, tangibly stored on a machine-readable medium, for dynamic access of master data, comprising instructions operable to cause a programmable processor to:

maintain a master database (Figure 1, reference numeral 116) having master data shared by at least two clients ([0050] page 8);

provide an interface for updating the master database ([0050] page 8);

provide an interface for mapping subsets of the master data into mapped data having a format that is acceptable to each client ([0050] page 8); and

provide a user interface for entering and displaying subsets of the master data ([0050] page 8).

INDEPENDENT CLAIM 42

Some embodiments of the Application are related to a computer program product, tangibly stored on a machine-readable medium, for dynamic access of master data, comprising instructions operable to cause a programmable processor to:

receive a first identifier used by a first client to identify a data object, and a request to delete the data object, the data object being stored in a database maintained by a data server, the database being accessible to the first client and a second client ([0051] page 9);

map the first identifier to a second identifier used by the second client to identify the data object ([0051] page 9);

map the first identifier to a third identifier used by the data server to identify the data object ([0051] page 9);

query the second client based on the second identifier to determine whether the second client is using the data object ([0051] page 9); and

if the second client is not using the data object, delete the data object from the database based on the third identifier ([0051] page 9).

INDEPENDENT CLAIM 43

Some embodiments of the Application are related to a computer program product, tangibly stored on a machine-readable medium, for dynamic access of master data, comprising instructions operable to cause a programmable processor to:

receive a first set of communications from a first client ([0052] page 9);

analyze the first set of communications to find a set of characteristics that the first client associates with a data object used in the first set of communications ([0052] page 9);

analyze other communications received from clients to find additional sets of characteristics that clients associate with data objects that are the same characteristics as the first set of characteristics ([0052] page 9);

place the first client and clients who sent a set of characteristics that are the same as the first set of characteristics into a client group ([0052] page 9); and

generate a data distribution path so that the programmable processor can route updates of the set of characteristics to the client group ([0052] page 9).

INDEPENDENT CLAIM 44

Some embodiments of the Application are related to a computer program product, tangibly stored on a machine-readable medium, for dynamic access of master data, comprising instructions operable to cause a programmable processor to:

associate master data with an object ([0053] pages 9-10);

send the master data to a master data server that stores master data associated with the object on a database ([0053] pages 9-10); and

access master data associated with objects on the database by requesting that an integration server that communicates with the programmable processor and the master data server map the data in the data server to a mapped data set that has a format conforming to rules defined by the programmable processor and send the mapped data set to the programmable processor ([0053] pages 9-10).

This summary does not provide an exhaustive or exclusive view of the present subject matter, and Appellant refers to each of the appended claims and its legal equivalents for a complete statement of the invention.

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-6, 8-11, 16, 19-23, 27-34, 40-42, 44-48, and 50 were rejected under 35 U.S.C. § 102(b) for anticipation by Bodamer et al. (U.S. 6,236,997).

Claims 39 and 43 were rejected under 35 U.S.C. § 102(b) for anticipation by Mahajan et al. (U.S. Patent No. 6,226,650).

7. ARGUMENT

A) The Applicable Law under 35 U.S.C. §102(b)

Under sub-section (b) of section 102, a person shall be entitled to a patent unless "the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States." 35 U.S.C. § 102(b).

A party asserting that a patent claim is anticipated under 35 U.S.C. § 102 (e) must show that "each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California,* 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). To anticipate a claim, a reference must disclose every element of the challenged claim and enable one skilled in the art to make the anticipating subject matter. *PPG Industries, Inc. V. Guardian Industries Corp.*, 75 F.3d 1558, 37 USPQ2d 1618 (Fed. Cir. 1996). The identical invention must be shown in as complete detail as is contained in the claim. *Richardson v. Suzuki Motor Co.,* 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim, but this is not an *ipsissimis verbis* test, i.e., identity of terminology is not required. *In re Bond,* 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

B) Overview of Bodamer et al. (U.S. 6,236,997)

Bodamer describes heterogeneous database environments that contain different, and often incompatible, types of database systems. These different database systems are typically purchased independently by businesses to serve a particular need, function or use of the data. As a result, businesses may have information spread across multiple database management systems. Since every database management system vendor attempts to provide a competitive edge over the offerings of its competitors, the different database management systems are almost by definition incompatible.[1] Bodamer then refers to a need for an arrangement that provides

[1] Bodamer, 1: 37-47.

scalable integration of foreign databases in a heterogeneous database environment, where operations necessary for execution of a client statement are selectively translated and sent to a foreign database system based on the corresponding capabilities of the foreign database system.[2] In response to this need, Bodamer proposes a system where an agent process provides an interface between foreign processes, executing operation requests sent by the agent process, and a local server process completing execution of a client statement based on results received by the agent process.[3]

An architecture for a database server process in a heterogeneous environment proposed by Bodamer is illustrated in FIG. 3A. Bodamer shows in FIG. 3A a client process 200 that supplies a statement such as a SQL statement to a local server 202. The local server 202 performs conventional query processing when processing native requests for data directly accessible by the local server 202. Where the client's statement requires an operation to be performed by a non-native (i.e., foreign) process, such as a foreign database server (FDS) 208, the local server 202 cannot complete execution of the statement without receiving the data from the foreign database server 208.[4] This situation is addressed in Bodamer by mapping a particular database operation to a target foreign process based upon the operation specified in the client statement and based upon definitions metadata for the heterogeneous services stored within a data dictionary stored by the local server 202.[5]

Bodamer explains that every relational database has its own set of data dictionary tables that store information (known as "metadata") about the objects in the database. This set of tables along with views defined on them are together called the "data dictionary." The local server 202 and the foreign database system may store similar metadata in their respective data dictionaries but the data in these two data dictionaries may be organized differently. For example, in the data dictionary maintained by the local server process 202, certain metadata is stored in a table entitled "user_catalog," while the foreign database 208 does not have a "user_catalog" table and stores similar metadata in two tables "sysusers" and "sysobjects." Bodamer, however, makes it

[2] Id., 2: 36-41.
[3] Id., 2: 49-54.
[4] Id., 4: 42-65.
[5] Id., 7: 9-14.

possible for a user to specify "user_catalog" as the target of the query and receive metadata stored in "sysusers" and "sysobjects" tables of the data dictionary maintained by the foreign database 208. The client 200 is thus given the appearance that the "user_catalog" table exists at the foreign database system 208. Bodamer provides an example of data dictionary translation where Oracle-compatible statement "select table_name, table_type from user_catalog @link" is the client statement "select table_name, table_type from user_catalog @link" to a Sybase-compatible query.[6]

C) Discussion of the rejection of claims 1, 19, 20, 28, 35, 40-42, and 44 under 35 U.S.C. § 102(b) as being anticipated by Bodamer et al. (U.S. 6,236,997)

As noted above, in order to anticipate a claim, a reference must disclose every element of the challenged claim,[7] where the elements are arranged as required by the claim.[8] Claim 1 requires a master database storing master data objects and a master data server using master identifiers to identify the master data objects. Claim 1 further requires an integration server operable to retrieve a master data object from the master database based on the master identifier in response to a request from a client to access master data identified by a client identifier. The relationship between the client identifier and the master identifier required by claim 1 is in that the integration server is to map the client identifier associated with the request from the client to the master identifier. Therefore, unless Bodamer discloses an integration server operable to respond to a request to access master data identified by a client identifier by retrieving a master data object from a master database based on a master identifier, where the master identifier is related to the client identifier through a mapping of the client identifier to the master identifier, Bodamer does not anticipate claim 1.

The techniques described in Bodamer, as discussed in grater detail above, are intended to address issues that arise in heterogeneous database environments that contain incompatible types of database systems.[9] An approach offered in Bodamer is a so-called data dictionary translation.

---

[6] Bodamer, 8: 10-67.
[7] *PPG Industries, Inc. V. Guardian Industries Corp.*, 75 F.3d 1558, 37 USPQ2d 1618 (Fed. Cir. 1996).
[8] *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).
[9] Bodamer, 1: 37-39.

Because data dictionaries of different database systems are organized differently, Bodamer proposes translating a query that is compatible with a local database system into a query compatible with a foreign database system. An example of such translation provided in Bodamer involves converting an Oracle-compatible query (that relies on metadata being stored in the "user_catalog" table) into a Sybase-compatible query. The translation process recognizes that while Oracle database system stores metadata in the "user_catalog" table, Sybase database system stores metadata in two tables - "sysusers" and "sysobjects."[10] The identifiers "sysusers" and "sysobjects" in the Sybase database system identify two respective tables in the Sybase data dictionary stored in the foreign database system 208 that controls a foreign database 308.[11] The tables identified by "sysusers" and "sysobjects" are stored in the foreign database system 208. Bodamer does not explain whether orhow the data dictionary or the tables in the foreign database system 208 identified by "sysusers" and "sysobjects" can be used in order to retrieve data stored in the foreign database 308.

Examiner cites the foreign database system (FDS) 208 illustrated in Figure 3A in Bodamer to show "a master data server" and a foreign database 308 to show "master database storing master data objects" recited in claim 1.[12] Examiner cites the local server 202 illustrated in Figure 3A in Bodamer to show "an integrated server" recited in claim 1.[13] In order to show "an integration server, in response to a request from a client to access master data identified by a client identifier, to map the client identifier to a master identifier, retrieve a master data object from the master database based on the master identifier, and map the master data object to a mapped data object based on a set of mapping rules associated with the client" recited in claim 1, Examiner refers to sections in Bodamer reproduced below.

> Each of these modules 210 in the heterogeneous services module 311 is configured to map a particular database operation to a target foreign process based upon the operation specified in the client statement and based upon metadata definitions metadata for the heterogeneous services stored within a data dictionary 220, described below. As shown

[10] Id., 8: 10-67.
[11] Id., 8: 32-35; 4: 60-67.
[12] Final Office Action, detailed Action, page 5, item 14.
[13] Id., page 6, item 14.

in FIG. 2B, the data dictionary 220 includes metadata definitions for use by the foundation services 204, as well as for the different modules 210 and 207.

(Bodamer, 7: 9-15.)

For example, assume the data dictionary for the local server process 202 includes a table entitled "user_catalog," and the client 200 issues a query on "user_catalog @ FDS," where "FDS" is an identifier for foreign database system 208. However, if the foreign database system 208 does not have the table "user_catalog," but rather the metadata is spread across different tables, the client 200 is given the appearance that the table exists at the foreign database system 208 in the structure perceived by the client.

Hence, the data dictionary translation gives the appearance to the client that the foreign database system 208 includes metadata in the format and structure as stored for the local server 202. Thus, the heterogeneous services modules 311 and 311' map the query to accommodate the data dictionary structures in the foreign database system 208, get the result back from the foreign database system 208, and return the results to the client 200 in the format of the local server 202.

(Bodamer, 8: 28-46.)

As is evident from the passages reproduced above, Bodamer refers to mapping a particular database operation to a target foreign process. (Bodamer, Fig. 3A; 7: 9-18.) Because an operation (or a process) is not an identifier and cannot be used as an identifier, an act of mapping of an operation to a process is distinct from mapping of one identifier to another identifier. While Bodamer refers to mapping a database operation to a foreign process[14], and mapping a query (by manipulating the syntax of the query) in order to accommodate data dictionary structures in a foreign database system, Bodamer is silent with respect to mapping a "client identifier" for master data to a "master identifier" that can be used to retrieve a master

[14] Bodamer, 7: 9-17.

data object from the master database, where "master identifiers" can be used "to identify the master data objects stored in the master database.

In the "Response to Arguments" section of the Final Office Action, Examiner refers to client query that is issued on "user_catalog@FDS" (attempting to access, at the foreign database system (FDS), metadata that corresponds to metadata stored in the "user_catalog" table at a local server). Examiner thus correlates "user_catalog" with "a client identifier." In an attempt to show the feature of "master identifier" recited in claim 1, Examiner states that distributed metadata at FDS is mapped to "user_catalog."[15] As explained above, the identifier "user_catalog" identifies a table in the local data dictionary, while distributed metadata referred to by Examiner is metadata that is spread across different tables (e.g., "sysusers" and "sysobjects") in the foreign data dictionary at FDS 208. There is no suggestion in Bodamer that the identifier "user_catalog" is mapped to metadata stored in multiple tables by the foreign database system 208. Therefore, metadata stored different tables in the foreign data dictionary at FDS 208 cannot be regarded as "master identifier" in the context of claim 1.

In the Advisory Action,[16] Examiner refers to the example of a data dictionary translation from an Oracle server to a Sybase server in Bodamer[17] and correctly points out that the parameter "user_catalog @link" from the original query is replaced with "susers@link SU, sysobjects@link SO." Although Examiner did not state that tables "sysusers" and "sysobjects" are intended as an example of distributed metadata that allegedly corresponds to the "master identifier" of claim 1, Applicants infer from Examiner's citing Bodamer at 8: 47-67 that "sysusers" and "sysobjects" are considered as possibly corresponding to the "master identifier" of claim 1. "Sysusers" and "sysobjects" in Bodamer identify two tables from a data dictionary maintained by a foreign database system and do not identify any particular data objects stored in a database 308 maintained by the foreign database system 208.

---

[15] Detailed Action, page 3, item 7.
[16] Advisory Action, continuation sheet. ("The metadata disclosed in Bodamer can be interpreted to include "a master identifier" as metadata is mapped to user_catalog.")
[17] Bodamer, 8: 47-67.

Because Examiner considers foreign database 308 to correspond to "a master database storing master data objects," "user_catalog" to correspond to "client identifier," table identifiers "sysusers" and "sysobjects" to correspond to "master identifier," and the local server 202 to correspond to the "integration server" of claim 1, in order to show an integration server to "retrieve a master data object from the master database based on the master identifier, in order to show that Bodamer anticipates claim 1, Examiner should be able to show that Bodamer discloses that the local server 202 is operable to retrieve a master data object from the database 308 based on the "sysusers" and "sysobjects." As explained above, in Bodamer, in response to a query directed to "user_catalog@FDS," the client may be provided with access to metadata stored in two data dictionary tables "sysusers" and "sysobjects" at the foreign database system 208 (not at the foreign database 308 cited by Examiner to show "master database" of claim 1). Therefore, the operation performed by heterogeneous services modules in Bodamer to access metadata stored in the foreign database system 208[18] does not correspond to an operation to "retrieve a master data object from the master database based on the master identifier," recited in claim 1. Furthermore, Examiner cites metadata stored in FDS 208 in Bodamer to show "the master data object from the master database" recited in claim 1.[19] Because Examiner cited the foreign database 308 to show "the master database" recited in claim 1, the metadata stored in FDS 208 (but not stored in the foreign database 308) does not read on the "the master data object from the master database" recited in claim 1 and therefore metadata obtained from FDS 208 cannot be correlated with a master data object retrieved from the master database. Therefore, Bodamer fails to disclose an integration server to "retrieve a master data object from the master database based on the master identifier," recited in claim 1.

Claim 1 further recites an integration server operable to "map the master data object to a mapped data object based on a set of mapping rules associated with the client." In order to show this feature, Examiner cites the example of a data dictionary translation from an Oracle server to a Sybase server in Bodamer[20]. As explained above, in Bodamer, metadata obtained in response to a query from the client is not retrieved from the database 308 (identified as corresponding to

[18] Id., 8: 41-46.
[19] Detailed Action, page 3, item 7.
[20] Bodamer, 8: 47-67.

"the master database" of claim 1 by Examiner). Furthermore, Examiner does not state what feature in Bodamer is considered as corresponding to "the master data object" retrieved from the master database, what feature in Bodamer is considered as corresponding to "a mapped data object," and what feature in Bodamer is considered as corresponding to "mapping rules associated with the client." Therefore, while Bodamer does not disclose retrieving a data object from the database 308 (correlated by Examiner to "the master database" of claim 1) based on the master identifier, there is no indication in Bodamer that any result of the query translated into a format of the foreign database is mapped to a mapped data object or that the client in Bodamer maintains any mapping rules that are to be applied to information retrieved in response to a query. Thus, Bodamer also fails to disclose or suggest an integration server to "map the master data object to a mapped data object based on a set of mapping rules associated with the client," recited in claim 1.

Because Bodamer fails to disclose every element of claim 1 in as complete detail as is contained in claim 1, claim 1 is patentable in view of Bodamer. It is respectfully requested that the rejection be reversed.

Claim 19 recites "an integration server, in response to a request from any one of the clients to access a master data object, to retrieve the master data object from the master database and map the master data object to a mapped data object based on a set of mapping rules associated with the client so that the mapped data object contains the subset of attributes in a format that can be processed by the client." Thus, claim 19 is patentable in view of Bodamer and should be allowed for at least the reasons articulated with respect to claim 1. It is respectfully requested that the rejection be reversed.

Claim 20 recites "mapping the master data object to a mapped data object based on a set of mapping rules associated with the client." Thus, claim 20 and its dependent claims are patentable in view of Bodamer and should be allowed for at least the reasons articulated with respect to claim 1. It is respectfully requested that the rejection be reversed.

Claim 28 recites "providing an interface for mapping subsets of the master data into mapped data having a format that is acceptable to each client." Examiner cites the following passage in Bodamer to show this feature.

> A fourth type of translation is data dictionary translation, executed in a SQL services module 210b within the heterogeneous services module 311 of the local server 202. Every relational database has its own set of data dictionary tables which store various kinds of information (known as "metadata") about the objects in the database created by its various users. This set of tables along with views defined on them are together called the "data dictionary." As a user makes modifications to his schema (for example, creating or deleting a table), the local server 202 will keep track of the modification by automatically adding or deleting entries in one or more tables of the data dictionary. (Bodamer, 8: 9-21.)

As is evident from the passage above, Bodamer is silent with respect to mapping one set of data to another set of data. Thus, Bodamer fails to disclose or suggest "providing an interface for mapping subsets of the master data into mapped data having a format that is acceptable to each client," as recited in claim 28. Thus, claim 28 and its dependent claims are patentable in view of Bodamer and should be allowed. It is respectfully requested that the rejection be reversed.

Claim 40 recites a processor to "map the master data object to a mapped data object based on a set of mapping rules associated with the client." Thus, claim 40 is patentable in view of Bodamer and should be allowed for at least the reasons articulated with respect to claim 1. It is respectfully requested that the rejection be reversed.

Claim 41 recites a processor to "provide an interface for mapping subsets of the master data into mapped data having a format that is acceptable to each client." Thus, claim 41 is patentable in view of Bodamer and should be allowed for at least the reasons articulated with respect to claim 1. It is respectfully requested that the rejection be reversed.

Claim 42 recites a processor to "map the first identifier to a second identifier used by the second client to identify the data object; map the first identifier to a third identifier used by the data server to identify the data object; query the second client based on the second identifier to determine whether the second client is using the data object." Thus, claim 42 is patentable in view of Bodamer and should be allowed for at least the reasons articulated with respect to claim 1. It is respectfully requested that the rejection be reversed.

Claim 44 recites a processor to "access the master data associated with the object on the database by requesting that an integration server that communicates with the programmable processor and the master data server map the master data in the master data server to a mapped data set that has a format conforming to rules defined by the programmable processor and send the mapped data set to the programmable processor." Thus, claim 44 and its dependent claims are patentable in view of Bodamer and should be allowed for at least the reasons articulated with respect to claim 1. It is respectfully requested that the rejection be reversed.

Claim 28 recites "providing an interface for mapping subsets of the master data into mapped data having a format that is acceptable to each client." Examiner cites the following passage in Bodamer to show this feature.

> A fourth type of translation is data dictionary translation, executed in a SQL services module 210b within the heterogeneous services module 311 of the local server 202. Every relational database has its own set of data dictionary tables which store various kinds of information (known as "metadata") about the objects in the database created by its various users. This set of tables along with views defined on them are together called the "data dictionary." As a user makes modifications to his schema (for example, creating or deleting a table), the local server 202 will keep track of the modification by automatically adding or deleting entries in one or more tables of the data dictionary. (Bodamer, 8: 9-21.)

As is evident from the passage above, Bodamer is silent with respect to mapping a set of data stored in the master database to another set of data. Thus, Bodamer fails to disclose or

suggest "providing an interface for mapping subsets of the master data into mapped data having a format that is acceptable to each client," as recited in claim 28. Thus, claim 28 and its dependent claims are patentable in view of Bodamer and should be allowed. It is respectfully requested that the rejection be reversed.

D) Discussion of the rejection of claims 39 and 43 under 35 U.S.C. § 102(b) as being anticipated by Mahajan (U.S. 6,226,650)

Mahajan discusses a system and method for updating client computer systems that enable client computer systems to be added to the ICDB system without requiring the ICDB system to create client-specific modification files for each client. In this system, continues Mahajan, *data on the server may be arranged in groups based on content and semantics*. One or more of the groups [of data] are assigned to each client depending on the data requirements of the client. Periodically, the server determines the data that has changed for each group [of data] since the last evaluation, and records those changes in a modification file. When a client connects to the server, it requests the modification files for the groups [of data] to which it subscribes, merges the downloaded modification files, filters the superfluous data, and updates its local database.[21]

It is submitted that assigning a group of data to a client, based on the data requirements of the client, is distinct from placing two clients into the same group. Thus, while Mahajan mentions arranging the data on the server into groups, Mahajan does not contemplate grouping clients based on respective sets of characteristics sent by the clients. In contrast, claim 39 recites "placing the first client and clients who sent a set of characteristics that are the same as the first set of characteristics into a client group; and generating a data distribution path to allow updates of the set of characteristics to be sent to the client group."

Furthermore, Mahajan does not mention, either in the above-identified passage or elsewhere in the specification, operations of "analyzing the first set of communications to find a set of characteristics that the first client associates with a data object used in the first set of communications" or "analyzing other communications received from clients to find additional

[21] Mahajan, 4: 15-29.

sets of characteristics that clients associate with data objects that have the same characteristics as the first set of characteristics" in order to place the first client and clients who sent a set of characteristics that are the same as the first set of characteristics into a client group, as recited in claim 39.

Because Mahajan fails to disclose every element of claim 39 in as complete detail as is contained in claim 39, claim 39 is patentable in view of Mahajan. It is respectfully requested that the rejection be reversed.

Claim 43 recites a processor to "place the first client and clients who sent a set of characteristics that are the same as the first set of characteristics into a client group; and generate a data distribution path so that the programmable processor can route updates of the set of characteristics to the client group" Thus, claim 43 is patentable in view of Mahajan and should be allowed for at least the reasons articulated with respect to claim 39. It is respectfully requested that the rejection be reversed.

SUMMARY

The reasons argued above are summarized as follows. First, Examiner failed to provide a reasoned rationale for anticipation of claims 1, 19, 20, 28, 35, 40-42, and 44 in view of Bodamer. Second, Examiner failed to provide a reasoned rationale for anticipation of claims 39 and 43 in view of Mahajan. For the reasons argued above, claims 1, 19, 20, 28, 35, 40-42, and 44 were not properly rejected under § 102(b) as being unpatentable over Bodamer et al, and claims 39 and 43 were not properly rejected under § 102(b) as being unpatentable over Mahajan.

It is respectfully submitted that the art cited does not render the above-identified claims anticipated and that the claims are patentable over the cited art. Reversal of the rejection and allowance of the pending claim are respectfully requested.

Respectfully submitted,

SCHWEGMAN, LUNDBERG & WOESSNER, P.A.
P.O. Box 2938
Minneapolis, MN 55402

Date December 29, 2008 By /Elena Dreszer/

Elena B. Dreszer
Reg. No. 55,128

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being filed using the USPTO's electronic filing system EFS-Web, and is addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on this 29 day of December 2008.

Dawn R. Shaw — /Dawn R. Shaw/

Name — Signature

8. CLAIMS APPENDIX

1. A system, comprising:

a master data server, to maintain a master database storing master data objects, the master data server using master identifiers to identify the master data objects, the master database being accessible to clients; and

an integration server, in response to a request from a client to access master data identified by a client identifier, to map the client identifier to a master identifier, retrieve a master data object from the master database based on the master identifier, and map the master data object to a mapped data object based on a set of mapping rules associated with the client.

2. The system of claim 1, wherein at least two clients use different client identifiers to identify a common master data object.

3. The system of claim 1, further comprising a mapping table to store information related to the mapping of the client identifiers to the master identifiers.

4. The system of claim 1, further comprising a mapping table to store mapping rules associated with the clients.

5. The system of claim 1, wherein the master data object has a plurality of attributes associated with characteristics of an entity represented by the master data object, and mapping the master data object to the mapped data object comprises retrieving a subset of the attributes from the master data object and formatting the subset of attributes based on rules defined by the client.

6. The system of claim 1, wherein the integration server dynamically maps the master data object in the master database to the mapped data object based on mapping rules defined by the

client each time the client requests for the master data without replicating the master data object at a database local to the client.

7. The method of claim 1, wherein the integration server comprises a cache to store master data objects that are requested by clients, and to provide stored master data objects to clients when the integration server receives requests that are identical to previous requests for access to the master data objects.

8. The system of claim 1, wherein the integration server comprises an exchange interface receives data that are published by a first client, and routes the published data to a second client that requested the published data.

9. The system of claim 8, wherein the integration server maps the data published by the first client to master data based on a first set of mapping rules associated with the first client, and maps the master data to mapped data that can be processed by the second client based on a second set of mapping rules associated with the second client.

10. The system of claim 1, wherein the integration server comprises a content integrator that finds characteristics that at least two clients associate with an object.

11. The system of claim 1, wherein the integration server comprises an adapter that receives communications from a client and extracts master data from the communications and forwards the extracted master data to the master data server.

12. The system of claim 1, wherein the master data server sends master data objects requested by clients to the clients without performing client authorization checks.

13. The system of claim 12, wherein the client performs authorization checks to limit access to the master data objects by users.

14. The system of claim 1, wherein the master data server performs authorization checks to limit access to the master data objects by users.

15. The system of claim 12, wherein the client performs authorization checks to limit access to the master data objects by processes running on the client.

16. The system of claim 1, wherein the master data server provides processes to allow the clients to modify the master data.

17. The system of claim 1 in which a portion of the master data objects are associated with products.

18. The system of claim 1 in which a portion of the master data objects are associated with business partners.

19. A system, comprising:

a master data server, to maintain a master database storing master data objects, each object having a set of attributes, the master database being accessible to clients, each client processing a subset of attributes of the master data objects; and

an integration server, in response to a request from any one of the clients to access a master data object, to retrieve the master data object from the master database and map the master data object to a mapped data object based on a set of mapping rules associated with the client so that the mapped data object contains the subset of attributes in a format that can be processed by the client.

20. A method, comprising:

maintaining a master database at a data server, the master database containing master data objects, the master database accessible to clients;

receiving a request from a client to access master data, the request containing a client identifier;

mapping the client identifier to a master identifier;

retrieving a master data object based on the master identifier;

mapping the master data object to a mapped data object based on a set of mapping rules associated with the client; and

sending the mapped data object to the client.

21. The method of claim 20, further comprising receiving a request from the client to modify the master data object to create a modified master data object, and querying the other clients to verify that the modified master data object conforms to consistency rules defined by the other clients.

22. The method of claim 21, further comprising, if a particular client does not respond to the query as to whether the modified master data object conforms to consistency rules defined by the particular client, placing the particular client on an exception list to indicate that the modified master data object has not been verified to conform with the set of consistency rules defined by the particular client.

23. The method of claim 22, further comprising, after a predefined period of time or when the particular client attempts to access data in the database, performing another attempt to verify whether the modified master data object conforms to the consistency rules defined by the particular client.

24. The method of claim 20, further comprising receiving a request from the client to delete the master data object from the master database, querying the other clients to verify that the master data object is not used by the other clients, and deleting the master data object from the master database after confirming the master data object is not used by the other clients.

25. The method of claim 20, further comprising storing a master data object in a cache, and retrieving the master data object from the cache rather than from the master database when a request for access to the master data object is identical to a previous request.

26. The method of claim 20, further comprising modifying the master data objects in the master database, and modifying the mapping rules to allow the clients to process modified master data objects without making modifications at the client.

27. The method of claim 20, wherein each master data object has attributes, each client processes a subset of the attributes, different clients process different subsets of the attributes, and the mapping rules associated with a client define which subset of attributes are processed by the client.

28. A method for maintaining data comprising:

providing a master database having master data shared by at least two clients;

providing an interface for updating the master database;

providing an interface for mapping subsets of the master data into mapped data having a format that is acceptable to each client; and

providing a user interface for entering and displaying subsets of the master data.

29. The method of claim 28, further comprising providing an exchange infrastructure that receives published data, published by a client, and routes the published data to another client that has requested the published data.

30. The method of claim 29, further comprising providing a content integrator to find characteristics that a first client and a second client commonly associate with an object.

31. The method of claim 30, further comprising receiving updates of the characteristics for an object from either one of the first and second clients, and sending the updates to the other of the first and second clients.

32. The method of claim 28, further comprising providing a content integrator to find characteristics that at least two clients associate with an object.

33. The method of claim 28, further comprising dynamically mapping the data in the master database to mapped data having a format conforming to rules defined by the client each time the client requests for data without replicating data stored in the master database to a database local to the client.

34. The method of claim 28, further comprising receiving updates of the characteristics for an object from either one of the first and second clients, and sending the updates to the other of the first and second clients.

35. A method for maintaining data, comprising:

receiving a first identifier used by a first client to identify a data object, and a request to delete the data object, the data object being stored in a database maintained by a data server, the database being accessible to the first client and a second client;

mapping the first identifier to a second identifier used by the second client to identify the data object;

mapping the first identifier to a third identifier used by the data server to identify the data object;

querying the second client based on the second identifier to determine whether the second client is using the data object; and

if the second client is not using the data object, deleting the data object from the database based on the third identifier.

36. The method of claim 35 in which querying the second client includes determining whether there is any reference to the data object in processes running on the second client and whether there is any reference to the data object in data buffers of the second client.

37. The method of claim 35 further comprising querying the second client to determine whether the second client objects to deletion of the data object, and preventing deletion of the data object if the second client objects.

38. The method of claim 35 in which the data object has a plurality of attributes, the first client configured to access a first subset of the attributes, the second client configured to access a second subset of the attributes, the second subset being different from the first subset.

39. A method comprising:

receiving a first set of communications from a first client;

analyzing the first set of communications to find a set of characteristics that the first client associates with a data object used in the first set of communications;

analyzing other communications received from clients to find additional sets of characteristics that clients associate with data objects that have the same characteristics as the first set of characteristics;

placing the first client and clients who sent a set of characteristics that are the same as the first set of characteristics into a client group; and

generating a data distribution path to allow updates of the set of characteristics to be sent to the client group.

40. A computer program product, tangibly stored on a machine-readable medium, for dynamic access of master data, comprising instructions operable to cause a programmable processor to:

maintain a master database at a data server, the master database containing master data objects, the master database accessible to clients;

receive a request from a client to access master data, the request containing a client identifier;

map the client identifier to a master identifier;

retrieve a master data object based on the master identifier;

map the master data object to a mapped data object based on a set of mapping rules associated with the client; and

send the mapped data object to the client.

41. A computer program product, tangibly stored on a machine-readable medium, for dynamic access of master data, comprising instructions operable to cause a programmable processor to:

maintain a master database having master data shared by at least two clients;

provide an interface for updating the master database;

provide an interface for mapping subsets of the master data into mapped data having a format that is acceptable to each client; and

provide a user interface for entering and displaying subsets of the master data.

42. A computer program product, tangibly stored on a machine-readable medium, for dynamic access of master data, comprising instructions operable to cause a programmable processor to:

receive a first identifier used by a first client to identify a data object, and a request to delete the data object, the data object being stored in a database maintained by a data server, the database being accessible to the first client and a second client;

map the first identifier to a second identifier used by the second client to identify the data object;

map the first identifier to a third identifier used by the data server to identify the data object;

query the second client based on the second identifier to determine whether the second client is using the data object; and

if the second client is not using the data object, delete the data object from the database based on the third identifier.

43. A computer program product, tangibly stored on a machine-readable medium, for dynamic access of master data, comprising instructions operable to cause a programmable processor to:

receive a first set of communications from a first client;

analyze the first set of communications to find a set of characteristics that the first client associates with a data object used in the first set of communications;

analyze other communications received from clients to find additional sets of characteristics that clients associate with data objects that are the same characteristics as the first set of characteristics;

place the first client and clients who sent a set of characteristics that are the same as the first set of characteristics into a client group; and

generate a data distribution path so that the programmable processor can route updates of the set of characteristics to the client group.

44. A computer program product, tangibly stored on a machine-readable medium, for dynamic access of master data, comprising instructions operable to cause a programmable processor to:

associate master data with an object;

send the master data to a master data server that stores master data associated with the object on a database; and

access master data associated with objects on the database by requesting that an integration server that communicates with the programmable processor and the master data server map the data in the data server to a mapped data set that has a format conforming to rules defined by the programmable processor and send the mapped data set to the programmable processor.

45. The computer program product of claim 44 wherein the integration server communicates with the programmable processor and the master data server dynamically.

46. The computer program product of claim 44 wherein the programmable processor sends a set of data to an exchange infrastructure that sends the set of data to another programmable processor that has requested the set of data.

47. The computer program product of claim 44 wherein the programmable processor sends characteristics that it associates with a data object to a content integrator which finds other programmable processors that associate the characteristics with the data object.

48. The computer program product of claim 44 wherein the programmable processor sends communications to an adapter that extracts master data from the communications and forwards the extracted master data to the master data server.

49. The computer program product of claim 44 wherein the master data server sends all data requested by the programmable processor to the programmable processor without performing programmable processor authorization checks.

50. The computer program product of claim 44 wherein the programmable processor can modify the master data stored in the master data server.

9. EVIDENCE APPENDIX

None.

10. RELATED PROCEEDINGS APPENDIX

No decision has been rendered with regard to the appeal pending with respect to patent application no. 10/365,672.